



iFlyBot-VLM Technical Report

Xin Nie^{1,†}, Zhiyuan Cheng^{1,†}, Yuan Zhang^{1,†}, Chao Ji², Jiajia Wu¹, Chenyu Xue¹, Jia Pan^{1,*}

¹iFlyTek Research and Development Group, ²LindenBot

Abstract

We introduce **iFlyBot-VLM**, a general-purpose Vision-Language Model (VLM) used to improved the domain of Embodied Intelligence. The central objective of **iFlyBot-VLM** is to bridge the cross-modal semantic gap between high-dimensional environmental perception and low-level robotic motion control. To this end, the model abstracts complex visual and spatial information into a body-agnostic and transferable "Operational Language", thereby enabling seamless perception-action closed-loop coordination across diverse robotic platforms. The architecture of **iFlyBot-VLM** is systematically designed to realize four key functional capabilities essential for embodied intelligence: 1) **Spatial Understanding and Metric Reasoning**; 2) **Interactive Target Grounding**; 3) **Action Abstraction and Control Parameter Generation**; 4) **Task Planning and Skill Sequencing**; We envision **iFlyBot-VLM** as a scalable and generalizable foundation model for embodied AI, facilitating the progression from specialized task-oriented systems toward generalist, cognitively capable agents. We conducted evaluations on 10 current mainstream embodied intelligence-related VLM (Visual-Language Model) benchmark datasets, such as Blink and Where2Place, and achieved optimal performance while preserving the model's general capabilities. We will publicly release both the training data and model weights to foster further research and development in the field of Embodied Intelligence.

Date: November 5, 2025

Correspondence: chaoji@iflytek.com, yuanzhang10@iflytek.com

Project Page: <https://xuwenjie401.github.io/iFlyBot-VLA.github.io/>

1 Introduction

In recent years, with the rapid development of LLM(Large Language Model) and large-scale pre-training technologies, the field of multimodal artificial intelligence has achieved milestone progress[1, 2]. In particular, Vision-Language Models (VLMs), leveraging their exceptional capabilities in image understanding, text reasoning, and cross-modal knowledge fusion, have become the core driving force behind general perceptual intelligence. These models have demonstrated performance surpassing traditional methods in various general perceptual tasks such as image-text recognition, visual question answering, and content generation. They have also partially integrated into human production and daily life, significantly expanding the application boundaries of AI.

Against this backdrop, researchers have naturally introduced the powerful semantic understanding and reasoning capabilities of VLMs into the fields of Embodied AI and robotics. The goal is to build Vision-Language-Action (VLA) models that can understand complex natural language instructions and execute

† denotes co-first author, * denotes the corresponding author

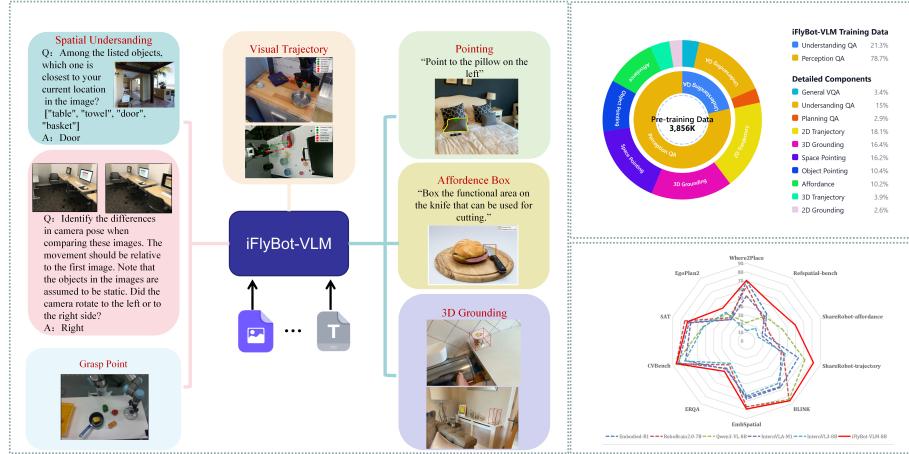


Figure 1 Overview. The model possesses capabilities in spatial object pointing, 2D trajectories, affordance regions, 3D bounding boxes (3Dbox), object grasping poses, object counting, spatial relationship judgment, multi-image mapping, and task planning. Additionally, it retains excellent multimodal capabilities such as caption generation, Grounding, and Optical Character Recognition (OCR). Moreover, it achieves state-of-the-art (SOTA) performance on multiple evaluation datasets.

physical tasks[3, 4]. Some cutting-edge works, such as π_0 [5] and $\pi_{0.5}$ [6], use VLMs as key modules to facilitate action generation, painting a promising prospect for the realization of general-purpose embodied agents. However, transferring the excellent performance of VLMs from the digital world to the physical world faces fundamental challenges. These VLA models often rely on massive, highly specialized training data to master a specific skill, which limits their application in diverse and open-ended environments. More seriously, once the operating environment, object posture, or background layout changes, their performance drops sharply. This phenomenon, known as "action memorization," makes them vulnerable to visual changes and results in extremely weak generalization ability.

On the one hand, this phenomenon is related to model architecture design, and on the other hand, is the huge gap between the current capabilities of VLMs and the capabilities required for embodied intelligence. As is well-known, robots, as intelligent agents operating in the physical world, their physical interactions involve not only digital information flows (such as images and text instructions) but also real-time processing of feedback from the physical world. This feedback includes self-state, fine-grained interactions, and complex dynamic and geometric constraints. The richness and uncertainty of the physical world lead to inherent flaws in the feature space constructed by VLMs in pure digital tasks. When dealing with 3D geometry and interaction relationships unique to the physical world, this feature space often cannot be directly and effectively mapped to robust physical action generation. Specifically, it is difficult for the model to correctly judge the front-back and up-down positions of objects from 2D images alone, to clearly distinguish environmental boundaries such as tabletops and walls, and even harder to accurately estimate the relative distance between objects—let alone plan collision-free operation trajectories. This indicates that existing VLMs cannot effectively extract high-dimensional physical features that are independent of the agent itself and essential for interacting with the physical world. Yet these capabilities are precisely the prerequisites for the generalization of robot manipulation.

Based on this, we propose the iFlyBot-VLM model, which possesses four core capabilities for the embodied domain: spatial understanding and measurement, interactive target localization, action abstraction and control parameter generation, and task planning. These capabilities enable the model to convert environmental information into "operation language" that robots can understand. This "operation language" is transferable and agent-independent, allowing for seamless bridging from perception to action. As illustrated in the Fig. 1,

the four capabilities are as follows:

- 1. Spatial Understanding and Measurement:** Equips the model with the ability to understand the spatial relationships between objects in the environment and estimate their relative positions.
- 2. Interactive Target Localization:** Provides multiple grounding methods, including 2D object detection, 3D object detection, object and spatial pointing, and object affordance region detection.
- 3. Action Abstraction and Control Parameter Generation:** Delivers graspable poses and operation trajectories that are most directly relevant to the manipulation domain.
- 4. Task Planning:** Can predict multi-step future atomic skills based on the current state, laying the foundation for the execution of long-horizon tasks.

This report will elaborate on the model from the perspectives of training data composition, model architecture, and experiments. Chapter 2 details the data composition and data production methods. Chapter 3 presents the model’s training framework and training strategies. Chapter 4 describes the evaluation dataset, evaluation metrics, and relevant experiments. Chapter 5 introduces related work, discussing the research progress in embodied foundation models. Chapter 6 provides a summary of this work and an outlook on future research directions.

2 The Model Architecture

As shown in Figure 2, iFlyBot-VLM inherits the robust three-stage "ViT-Projector-LLM" paradigm commonly used in current mainstream VLMs. Specifically, the model achieves efficient feature alignment through a dedicated Vision Transformer (ViT) trained via incremental pre-training, an advanced LLM, and a concise, randomly initialized multi-layer perceptron (MLP Projector). Our model is partially inspired by the architectural design of InternVL3[7]. The LLM and MLP Projector components of our model are initialized using the parameters from InternVL3-8B.

The core innovation of iFlyBot-VLM lies in the improvement of the positional encoding (PE) layer of ViT. Unlike relying solely on the original 448-dimensional positional encoding, this study adopts the bicubic interpolation method to upsample the learned positional embeddings from 448 dimensions to 896 dimensions, thereby forming a new positional encoding mechanism—Dimension-Expanded Position Embedding (DEPE). This innovative method provides a more refined spatial context vector for each visual token, enabling the model to capture more complex positional information and relative spatial relationships without increasing the sequence length. As a result, the model demonstrates superior performance and generalization capabilities in tasks such as fine-grained visual reasoning and precise localization.

3 Training Data

We train the iFlyBot-VLM model on a mixture of data sources, which totals approximately 380W samples, covering carefully crafted interactive grounding data, action control parameters, task planning data, and spatial understanding data. Among them, the interactive grounding data amounts to about 215W samples; the action control parameter data—composed of 2D trajectories and object grasping points—totals 73.7W samples; the spatial understanding data sums up to 57.8W samples; and to prevent model forgetting, 13W high-quality general VQA (Visual Question Answering) samples have been added.

3.1 Spatial Understanding Data

Existing Vision-Language Models (VLMs) are capable of performing high-level reasoning and scene interpretation, yet challenges persist in semantic reasoning and scene understanding within robotic manipulation scenarios. As shown in the Figure 5 is an overview of our spatial understanding data. The image-text corpora used for VLM pre-training lack high-quality, fine-grained spatial information that serves as a prerequisite for robots to recognize long-tail objects, interpret complex scenes, reason about spatial relationships, and plan physical interactions. Embodied spatial understanding focuses on a robot’s comprehension of object geometry, spatial relationships between objects, object counting, relative depth, camera motion perception,

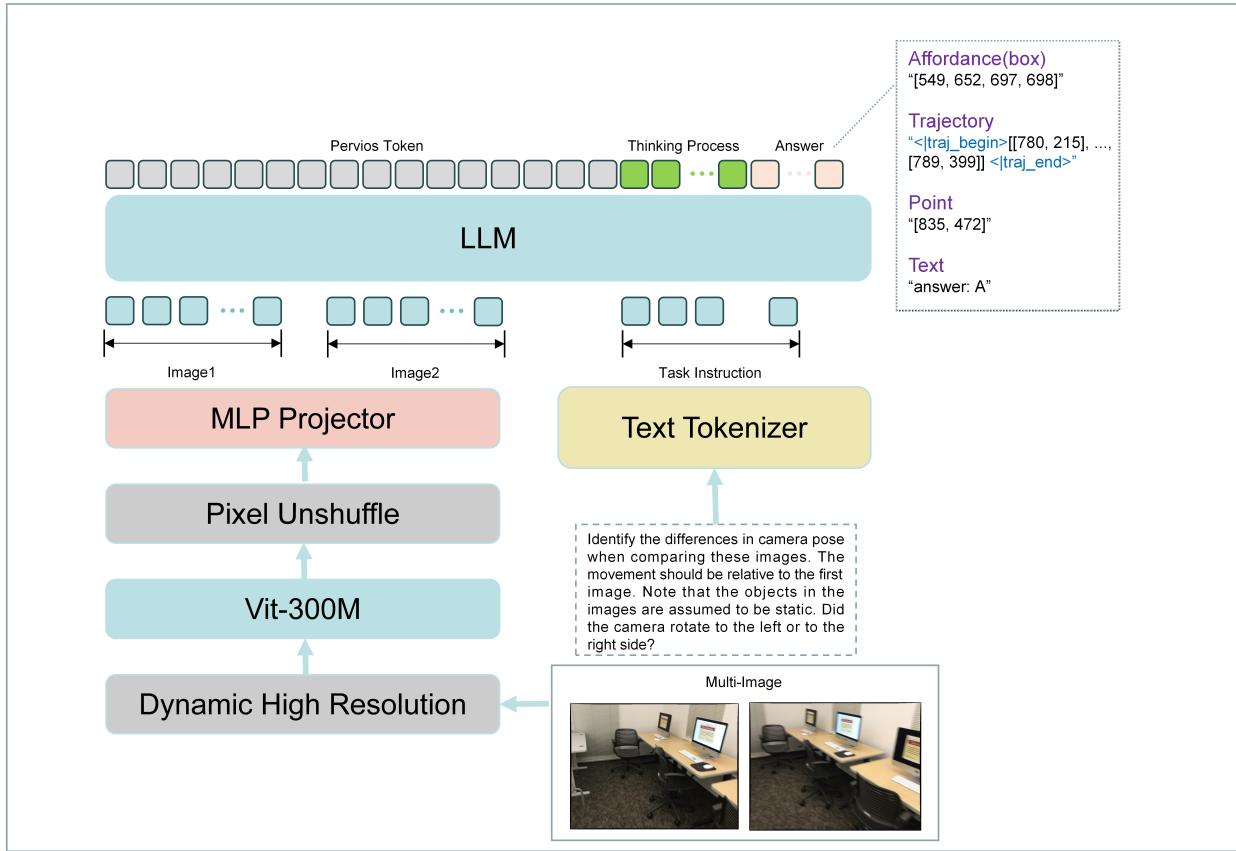


Figure 2 The iFlyBot-VLM Model. iFlyBot-VLM is a three-stage "ViT-Projector-LLM" paradigm from established Vision-Language Models.

and cross-visual spatial layouts. It covers a range from low-level pattern matching (e.g., visual correspondence) to high-level spatial reasoning (e.g., relative depth).

3.1.1 Visual Correspondence Data

This data is designed to enhance the ability of MLLMs to understand and identify the same scene point across different viewpoints, lighting conditions, or timeframes. Using ScanNet[8] and the method outlined in MultiSPA[9], we generated 500K pieces of qualitative data. Given two images and a pixel position in the first image, the model must qualitatively identify the corresponding pixel in the second image that shares the same 3D position. Visual annotations were added to both images: a position point was marked in the first image, and the corresponding position point was marked in the second image via projection. Three additional position points were also annotated to form a multiple-choice question, with the model tasked with selecting the correct annotation.

3.1.2 Relative Depth Data

This data aims to improve the geometric understanding capabilities of VLMs. We carefully selected datasets using task data from robotic scenarios (e.g., AgiBotWorld[10], RoboMind[11]), employed DepthAnything[12] for depth fitting, selected multiple position points in images, and annotated them using depth maps. This process ultimately yielded 34K pieces of high-quality multiple-choice data. To enhance the accuracy of robots in task execution, we designed two types of relative depth data: one for qualitative judgment of relative depth from the current first-person perspective to the camera lens, and another for qualitative judgment of relative depth from the third-person perspective to the robot's gripper.

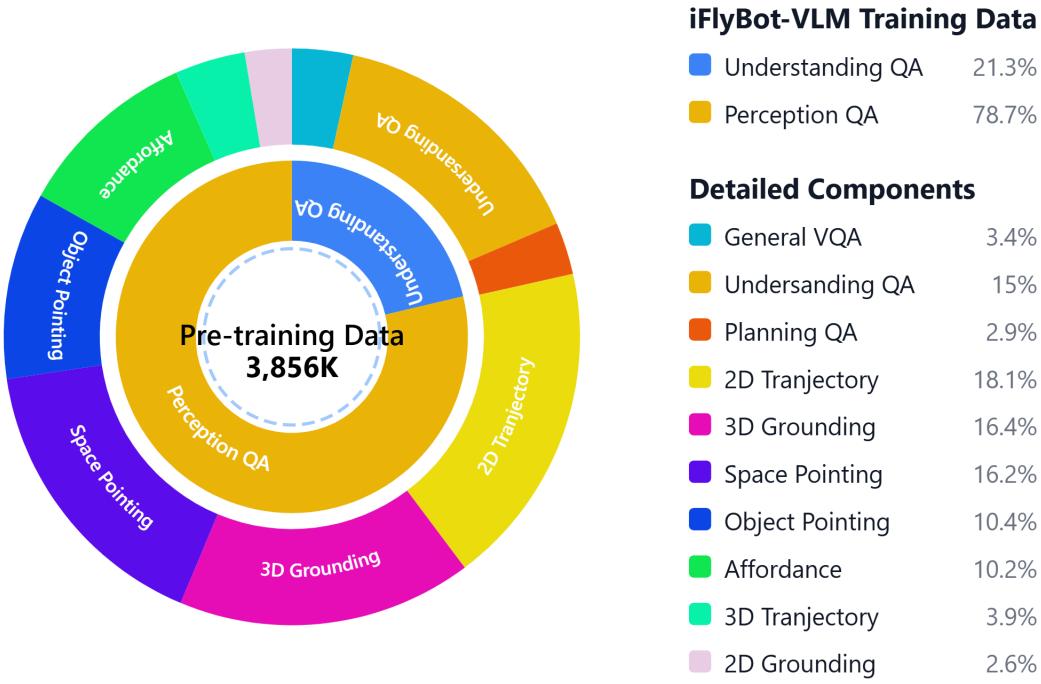


Figure 3 Data distribution chart, covering 13 subcategories of data.

3.1.3 Object Counting Data

This data is intended to strengthen the detection, recognition, and compositional reasoning capabilities of VLMs for complex scenes where objects may overlap, be occluded, or vary in size and shape. Since pre-trained models already possess a certain level of object counting ability, we used the object counting segment of PixMo[13] data and TallyQA[14] data, resulting in a total of 84K data points.

3.1.4 Camera Motion Perception Data

This data is used to boost the multi-view reasoning capabilities of Multimodal LLMs. Given two images, the model estimates the camera's relative motion from the first viewpoint to the second, including translation and direction. Using ScanNet[8] and the method described in MultiSPA[9], we annotated rotations as "left" or "right" based on yaw angles, and defined translation directions as "left" or "right" according to the y-axis. To avoid ambiguity during model training, we performed more precise data annotation—for example, removing data where rotation and translation contradicted each other, and appropriately increasing translation distance and rotation angle. Through high-precision annotation, this segment of data totals 61K points.

3.1.5 Object Spatial Relationship Data

This data is designed to help VLMs better understand spatial relationships between objects in a scene and improve robots' ability to interpret complex visual environments. In this dataset, we designed a rich set of spatial relationships, including "left", "right", "in front of", "behind", "above", "under", "between", "nearest", "farthest", "center", and "edge". When designing question-and-answer (Q&A) data, we created multiple types (e.g., single-choice, multiple-choice, open-ended Q&A, and image captions) to ensure the model fully learns spatial relationships. Our data sources include the OpenImage[15] and VisualGenome[16]. Through high-precision annotation, this dataset contains 406K data points in total.

To further enhance the model's spatial understanding capabilities, we designed third-person perspective data, which requires judging the relative position of objects from the viewpoint of another observer in the image.

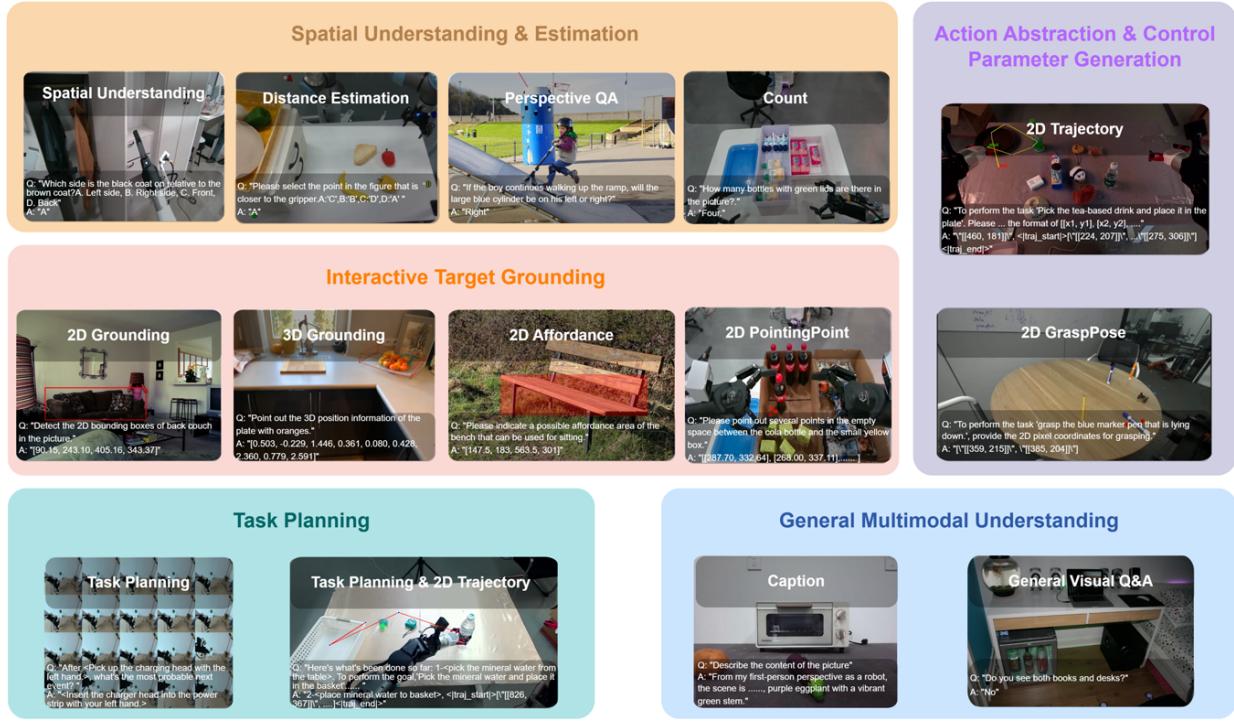


Figure 4 Training Data. The training data presentation covers five categories of data, including General Multimodal Understanding, Action Abstraction & Control Parameter Generation, Spatial Understanding, Interactive Target Grounding, and Task Planning.

An example question is: "If you were the person on the bicycle, facing the same direction as him, would the white truck be in front of or behind you?" This type of data requires prominent targets in the image. Using the YOLO11n[17] algorithm, we filtered images meeting the requirements from the VisualGenome[16], and used Gemini-2.5[18] to generate the data—resulting in 3K high-precision data points.

3.2 Spatial Perception Data

In robotic manipulation tasks, in addition to equipping the model with spatial understanding capabilities, precise spatial perception capabilities are also crucial. This involves the position of the manipulated object, the operable regions on the manipulated object, the end-effector posture when the object is manipulated, the positions where the object can be placed, and the trajectory of the end-effector during the object manipulation process.

3.2.1 2D Grounding Data

We prepared the iFlyBot-2dGrounding dataset, which includes general scenarios and robotic manipulation scenarios. The general scenario dataset is derived from object detection data in RefCOCO[19] and RoboPoint[20]. The objects involved in these datasets include furniture, clothing, food, humans, buildings, plants, animals, etc., and the scenarios cover streets, homes, stadiums, seas, etc. We filtered out data with ambiguous references or meaningless references in the object detection boxes to form the high-quality iFlyBot-2dGrounding-General-250K dataset. However, the scenarios and referenced objects in this data are too generalized, so their role in robotic manipulation tasks is not very direct. Based on this, we supplemented data for robotic manipulation scenarios. This supplementary data is sourced from AgibotWorld[10], RoboMind[11], and BridgeData[21]. We used Gemini-2.5 and the object lists in the datasets to identify the objects present in the scenarios, then used GroundingDINO[22] to obtain their 2D bounding boxes (2Dbox). After filtering, we constructed the high-quality iFlyBot-2dGrounding-RobotScene-500K dataset.

| | | |
|---|---|---|
| <p>Visual Correspondence</p>  <p>Q: Find the points that correspond between the images. Which point in Image-2 is the same as the circled point in Image-1? A: B</p> | <p>Relative Depth</p>  <p>Q: Which of the following points is closest to the left gripper? A: A</p> | <p>Counting</p>  <p>Q: "How many chairs? Please answer directly with only the corresponding number and nothing else." A: 2</p> |
| <p>Camera Movement</p>  <p>Q: Identify the differences in camera pose when comparing these images. The movement should be relative to the first image. Note that the objects in the images are assumed to be static. Did the camera rotate to the left or to the right side? A: left. B: right.</p> | <p>Spatial Relation</p>  <p>Q: Which of the following statements about the positions of the objects in the image is correct? A. The crumpled paper is on the right side of the white bottle, B. The trash can is in front of the dustpan, C. The white bottle is on the left side of the trash can, D. The crumpled paper is in front of the trash can. A: D</p> | <p>Perspective</p>  <p>Q: If you were standing in the lady's position, facing the same direction as her, would the two people walking on the path be on your left or right? A: Left</p> |

Figure 5 Spatial Understanding Data. This figure presents partial examples of spatial understanding data, including visual correspondence, relative depth, counting, camera movement, spatial relation and perspective

3.2.2 2D Pointing Data

For pointing points, we divided them into two categories: object-pointing datasets and placement-position-pointing datasets. Both categories use 2D points [x, y] as reference results. To expose the model to more diverse data features and distributions, help it learn more general patterns and rules, and thereby improve its generalization ability, we set the reference results as random point sets containing 1 to 9 points. We introduced the RefSpatial[23] dataset, which includes 2.5 million high-quality examples and 20 million question-answer (QA) pairs. It contains fine-grained annotations ranging from broad categories to precise spatial references, covering 31 types of spatial relationships. The dataset is composed of multi-source data, integrating 2D web images, 3D embodied videos, and simulated scenarios. This dataset includes both object-pointing data and free-region-pointing data. After filtering the 3D embodied data and simulator data in it, we split the data by type to form two datasets: iFlytek-2dPoints-Location-RefSpatial-270W (for object pointing) and iFlyBot-2dPoints-Placement-RefSpatial-380W (for free-region pointing). In addition, we introduced the RoboPoint[20] dataset, which covers 770K pixel point prediction tasks. The answers are mostly represented as lists of 2D points, indicating positions on the image (e.g., task prompts like "locate the object between the markers"). This dataset also includes two categories: object pointing and placement-position pointing. Notably, the images in this dataset typically have a manually labeled reference box for guidance.

For the object-pointing dataset, we specifically introduced RoboReflect-36K[24]. Most of the aforementioned datasets focus on indoor home scenes or desktop scenes, lacking robotic arm manipulation scenarios. To address this, we used datasets including AgibotWorld[10], Droid[25], RoboMind[11], BridgeData[21], and our self-collected iFlytekPickAndPlace data to create the object-pointing dataset for desktop manipulation scenarios: iFlyBot-2dPoints-Object-RobotScene-143K. The creation process was as follows: first, we obtained object bounding boxes using the same method as for iFlyBot-2dGrounding-RobotScene-500K; then, we

extracted object contours via SAM (Segment Anything Model); finally, we randomly sampled 1–9 points within these contours. During the creation of this dataset, we defined two types of prompts:

- Direct prompts that explicitly specify the object to point to, e.g., "Please indicate the position of the apple";
- Enhanced prompts that indirectly specify the target object using spatial orientations to further improve the model's spatial understanding ability, e.g., "Please indicate the third object from the left on the front side of the basket on the table".

To further strengthen the model's understanding of relative distance in spatial comprehension, we introduced the Arkit[26] dataset. This dataset labels the 3D bounding box coordinates of objects, where the 3D positions and spatial relationships of objects are fixed. We randomly selected objects and used Gemini-2.5 to generate deterministic spatial descriptions between them (e.g., "The coffee table 0.8 meters to the right of the bed"), resulting in the dataset iFlyBot-2dPoints-Location-Arkit-13.9K.

For the object placement position dataset, we also supplemented data for desktop manipulation scenarios. We used robotic manipulation datasets including Droid[25], FSD[27], and our self-collected IflytekPickAndPlace data, from which we filtered out only pick-and-place task data. This data typically consists of real-time joint and gripper state variables, and operation videos captured by the head camera. We used the last frame of the fixed head camera sequence as the image for the dataset to be created. For answers composed of point sets, the creation steps were:

- First, we determined the key frames of gripper opening and closing based on changes in gripper state variables. We then used the 2D coordinates of the gripper's fingertip center at these key frames as a prior;
- Next, we obtained the object's contour via GroundingSAM[28];
- Finally, we randomly generated point sets within the contour to form the answers.

For prompts, we used either the frame before the object was picked up or the first frame of the sequence as the reference image, and used Gemini-2.5 to generate spatial descriptions based on the object at the object point. We defined three types of spatial descriptions:

- Between objects;
- Between an object and the observer;
- Between an object and a specific side of the tabletop.

In this way, we constructed the placement-pointing dataset for desktop manipulation scenarios: iFlyBot-2dPoints-Placement-RobotScene-24K. We also reutilized the Arkit[26] dataset to generate placement-region-pointing data: iFlyBot-2dPoints-Placement-RobotScene-12K.

3.2.3 2D Affordance Data

An object's affordance region refers to the area of the object that can be manipulated. We categorize its references into two types:

- Object-centric references: These refer to the regions of the object that can be "manipulated," where "manipulation" includes actions such as sitting on, tapping, picking up, opening, and cutting.
- Robot-centric references: These refer to the specific regions of a target object that can be manipulated when the robot intends to operate on that object.

We use the coordinate format [x1,y1,x2,y2] to represent these regions. We first introduced part-level affordance datasets (HANDL[29], InstructPart[30], rgbd-part[31], and iit-aff[32]) to create 231.3K data samples. These datasets describe the functional roles of specific parts of objects, with examples such as "Which region of the cabinet can be pulled open?" or "Which part of the axe can be picked up?"

Next, we introduced the PACO[33] dataset, which covers 75 object categories, 456 object part categories, and 55 attributes, spanning both image (LVIS[34]) and video (Ego4D[35]) data sources. We filtered this dataset based on criteria such as image resolution and bounding box size, resulting in the iFlyBot-2dAfford-PACO-160K dataset.

Finally, we introduced the affordance dataset AGD20K[36]. For images in its training set, we generated prompts using Gemini-2.5, obtained affordance regions via GroundingSAM[28], and refined these regions through manual adjustments. We then calculated the minimum enclosing rectangles for these regions to form the iFlyBot-2dAfford-AGD20K-12K dataset.

3.2.4 2D GraspPose Data

The graspable pose of an object refers to the position and posture of the end-effector when manipulating the object. We use a two-finger gripper as the end-effector, and the grasp pose only needs to be represented by 2 points, i.e., $[[x_1, y_1], [x_2, y_2]]$. The line connecting these two points can be used to indicate the grasp posture.

We used robotic datasets (AgibotWorld[10], Droid[25], RoboMind[11], OpenX-Embodiment[37]) to extract key frames of objects before manipulation—specifically, the image frame captured at the mid-moment when the gripper transitions from an open to a closed state. A self-trained object detection model, Gripper-Finger-Detector, was used to detect the two fingertip points, and these two points in the image coordinate system were taken as the grasp points.

Gripper-Finger-Detector was developed through the following process: 20K images were selected and filtered from the aforementioned robotic datasets based on gripper type diversity and scenario diversity, covering 10 gripper styles. After manual annotation, these images were used to train the object detection model. Using this method, we created the dataset iFlyBot-2dGraspPose-OpenSourceRobot-32K from the AgibotWorld[10], Droid[25], RoboMind[11], OpenX-Embodiment[37] datasets.

For the self-collected IflytekPickAndPlace data, we have accurate structural dimensions and camera intrinsic/extrinsic parameters. First, we obtained the key grasp frames based on joint state information and gripper information. Next, we calculated the 3D pose of the two gripper fingertips in the camera coordinate system. Finally, we projected these 3D coordinates onto the 2D image via intrinsic parameters to obtain the grasp point coordinates, thus forming the dataset iFlyBot-2dGraspPose-IflytekPickAndPlace-60K.

3.2.5 2D Trajectory Data

We define an object’s operable trajectory as the operation trajectory of the end-effector. We selected 30 sub-actions as atomic skills, including pick, place, open, close, grasp, put, move, pull, push, etc., and defined boundaries for these atomic skills. For example, "pick" refers to the process where the gripper transitions from an empty state (initial state) to grasping an object and then lifting it a certain distance; "place" refers to the process where the gripper (with a load, in a suspended state) places the object and then returns to an empty suspended state.

We introduced the LLARVA[38] dataset and first filtered images with both width and height greater than 224. Additionally, through random sampling checks, we removed images with chaotic scenes or unclear operation subjects. For the point sets in the dataset, we filtered out invalid trajectory data for movement-related instructions (e.g., pick, place): specifically, data where the total path length significantly differed from the total displacement, or where the total path length was excessively short. Since the trajectory points in this dataset were relatively dense, we resampled the trajectories into random sequences of 6 to 8 points by retaining key points. Finally, we modified the dataset’s prompts according to the rules of our defined atomic skills, resulting in the dataset iFlyBot-2dTrajectory-LLARVA-560K.

To expand the scenario coverage, we also created trajectory data using open-source datasets (AgibotWorld[10], Droid[25], RoboMind[11], BridgeData[21]). The process was as follows: first, we identified key frames of gripper opening/closing based on joint states, and retained 50 sequence frames (including these key frames) for each sequence. Next, we used our self-trained object detector (Gripper-Finger-Detector) to detect the gripper in each sequence frame, and took its central area as trajectory points. Taking Droid data as an

example: we first split sequences into sub-task segments based on the key frames of gripper opening/closing; then, using Gemini-2.5[18], we split the prompts for these sub-task segments according to our defined list of atomic skills and the sequence descriptions in the original dataset; for each sub-task segment, we extracted the corresponding point set from the original trajectory points; after filtering the point sets based on criteria such as object detection thresholds and the ratio of path length to displacement between the start and end points, we resampled the trajectory sequence for each atomic skill into 6 to 8 random points—with special retention of key frames (the start frame of the sequence and the opening/closing key frames). This process formed the dataset iFlyBot-2dTrajectory-OpenRobotDatasets-181K.

Using the same method as for fingertip points, we obtained real-time pixel coordinates of the gripper center via 3D-to-2D projection from our self-collected IflytekPickAndPlace dataset. We split sequences according to the boundaries of custom atomic skills, extracted trajectory points for each atomic skill, and then resampled them to form the dataset iFlyBot-2dTrajectory-IflytekPickAndPlace-120K. Notably, in addition to trajectories for single atomic skills, we also created two-step trajectory data (e.g., "pick the apple and place it into the basket"). The number of points in such trajectories was limited to 8, including points from key frames (opening/closing frames) and the start frame.

To equip the model with reflective capabilities, we created negative examples for the trajectory data. We applied six types of perturbations to valid trajectories to render them irrational:

1. Perturb the first half of the trajectory, so that the start point of the trajectory is not at the gripper;
2. Perturb the second half of the trajectory, so that the end point of the trajectory is not near the target object;
3. Perturb the middle part of the trajectory, so that the middle segment passes through obstacles;
4. Mismatch between text and image: the image lacks the object to be manipulated;
5. Trajectory missing: the first half of the trajectory is incomplete;
6. Trajectory missing: the second half of the trajectory is incomplete.

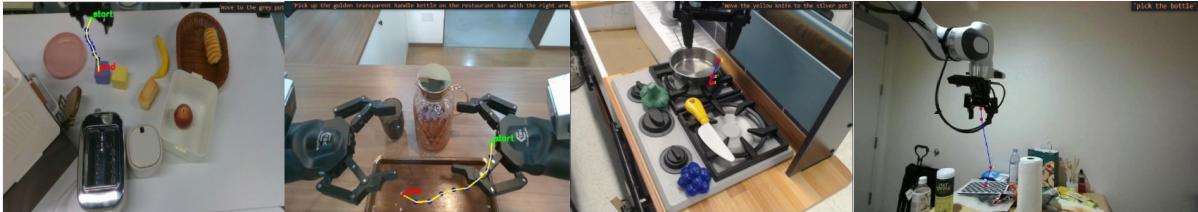


Figure 6 Trajectory negative example data.

The data structure was designed as follows: the question provides the sequence of trajectory points for the (perturbed) trajectory, and the answer includes a judgment on the trajectory's rationality and the reason for its irrationality .

In addition, we created a batch of pseudo-reinforcement learning data: the question presents different trajectories, and the answer describes the outcome of following each trajectory (e.g., "the apple was picked up", "the water bottle on the table was touched", "the banana was placed outside the plate"). For this process, we constructed a virtual environment and used human-annotated data as feedback, forming the dataset iFlyBot-2dTrajectory-IflytekPickAndPlace-Reflection-120K.

3.2.6 3D Grounding Data

To enhance the model's spatial measurement capabilities, we created a dataset for 3D bounding boxes (3D boxes), where each 3D box is represented by 9 numerical values: [cx, cy, cz, L, W, H, roll, pitch, yaw]. The first three values (cx, cy, cz) denote the position of the object's centroid in the camera coordinate system. L,

W , and H represent the object's length, width, and height respectively, with units in meters (m). Roll, pitch, and yaw indicate the object's orientation, with units in radians (rad).

We introduced the CA-1M[39] dataset. Based on the original dataset, we sampled across different shooting scenarios to ensure diversity, used Gemini-2.5[18] to identify the correct name of each object in the scene, and filtered the dataset based on object size, visibility, and scenarios with mutual occlusion between multiple objects—all to improve the overall quality of the dataset.

For the SUNRGBD[40], Objectron[41], and ARKitScenes[26] datasets, we performed filtering based on object visibility and 2D bounding boxes to ensure that all annotated objects in the scene are visually visible.

We noticed that existing open-source datasets contain scenarios with multiple objects that have identical shapes or labels. For such objects, we uniformly added directional ordinal descriptions (e.g., "the first cup from the left") to ensure that each label in the scene refers to a unique object. This enables the model to better locate and reference target objects in complex scenes through directional information.

Furthermore, in addition to direct questions about an object's 3D box (referred to as Direct-QA), we also created more challenging referential 3D box question-answer (QA) data (referred to as Refer-QA). Instead of directly asking for an object's 3D box, we ask for the 3D box of an object located in a specific direction relative to another object. This simulates the real-world interaction logic where humans often locate targets through "relative positions" in complex scenes, further enhancing the model's robustness and practical value when facing multi-object interference and needing to locate objects based on spatial correlations.

3.3 Spatial Perception Data with COT

For complex tasks—such as pointing problems involving spatial understanding, part-level affordance judgment, and operation trajectory planning—the core challenge lies in the need for multi-step logical deduction and decomposition of spatial relationships. Directly generating final results can easily lead to jumpy errors or overlooked key constraints. Therefore, we introduce chain-of-thought (CoT) to guide the model in step-by-step reasoning. By breaking down complex tasks into coherent stages such as "identify targets → analyze constraints → derive intermediate steps → verify rationality," the model progresses incrementally based on previous reasoning results. This not only ensures the interpretability of the reasoning process but also reduces omissions caused by one-step generation, ultimately improving the processing accuracy and reliability of complex tasks. Taking trajectory data as an example, we use four steps to guide the model's reasoning:

- Parsing the initial state of objects and the gripper, as well as the placement position;
- Clarifying the task and planning obstacle avoidance paths;
- Describing the complete trajectory step-by-step (including details of gripper movements);
- Drawing the final trajectory conclusion.

In this process, the model is prompted to first output the positions of objects and the gripper, which essentially trains the model to master object pointing and target detection capabilities before further deriving trajectories based on established key points. Using this method, we constructed 5K data samples, where the thinking process is enclosed between `<think>` and `</think>` as delimiters, followed by the trajectory prediction result.

To enhance the model's ability to indicate object functional regions, we designed a thinking chain for data augmentation based on the AGD20K[36] and InstructPart[30] datasets. This chain includes identifying user intent, locating object poses, activating the model's prior knowledge, and positioning functional regions. During the generation of augmented data, we used Grounding DINO to assist in object localization, overcoming potential positioning or reference errors that might occur during generation.

3.4 Task Planning Data

For long-horizon tasks, we often require the model to decompose the task into subtasks for execution, and we incorporate human operation data to improve task accuracy. We utilized the training data from EgoPlan

Table 1 Model Performance on Benchmarks about Spatial Perception

| Models/Metrics | Where2Place | Refspatial-bench | ShareRobot-affordance | ShareRobot-trajectory |
|---------------------|-------------|------------------|-----------------------|-----------------------|
| Embodied-R1[44] | 69.5 | 39.5 | 22.69 | 63.29 |
| RoboBrain2.0-7B[45] | 63.59 | 32.5 | 28.05 | 44.888 |
| Qwen3-VL-8B[46] | 21.2 | 35.75 | 43.72 | 71.21 |
| InternVLA-M1[47] | 52.19 | 35.5 | 18.99 | 42.33 |
| InternVL3-8B[7] | 12 | 18 | 14.45 | 47 |
| IflyBot-VLM-8B | 70.23 | 51.5 | 59.61 | 81.74 |

Bench[42], where the video data is sourced from Epic-Kitchen[43] and Ego4D[35]. These videos record real first-person perspective data of daily human activities.

Furthermore, to deepen the model’s understanding of task decomposition and how to execute each subtask, we created a hierarchical task trajectory dataset. This dataset integrates both temporal and perceptual data—when outputting subtasks, it simultaneously outputs trajectory results. Based on this, we used our self-collected IflytekPickAndPlace dataset: we selected data with more than 2 steps, split the trajectories using grasp key frames, and used Gemini-2.5[18] to generate prompts for each segment. Example question descriptions generated include: "For the task of putting all objects on the table into the plate, please predict the remaining steps and the trajectory for each step."

4 Experiments

To evaluate the model’s performance, we selected commonly used benchmark datasets in the industry.

4.1 Spatial Perceptive Result

4.1.1 For Pointing Point Evaluation

First, we chose the Where2Place[20] benchmark, an indoor scene dataset focused on object placement regions. It includes 70 seen data samples and 30 unseen data samples. Additionally, we introduced RefSpatial-bench[24], a comprehensive benchmark consisting of 100 object-pointing data samples, 100 placement-region-pointing data samples, and 70 unseen data samples. Both datasets use mask regions to represent ground truth regions. The evaluation score for a single image is calculated as the ratio of the total number of correct predicted points to the total number of predicted points. The evaluation score for a single image is calculated as the ratio of the total number of correct predicted points to the total number of predicted points. Let N_{correct} denote the total number of **correct predicted points** (i.e., the predicted points that fall into the predefined target region, such as a mask area). Let N_{total} denote the **total number of predicted points** generated for the single image. The evaluation score S for the single image is specifically calculated by the following formula:

$$S = \frac{N_{\text{correct}}}{N_{\text{total}}}$$

The total score for the dataset is the average of the sum of scores across all single images. It can be observed that despite being trained using the SFT method, our model outperforms all evaluated models on Where2Place, achieving a score of 70.3. In contrast, RoboBrain2.0-7B scores 63.59, and Embodied-R1 that trained with reinforcement learning scores 69.5. Partial results are shown in Table 1. Partial results are presented in Appendix B. On Refspatial-bench, We also ranked first. We achieved a score of 51.5, which is 12 points higher than that of Embodied-R1.

4.1.2 For Affordance Dataset Evaluation

We used the affordance data from Sharerobot-bench[48] for testing. Before testing, we identified unreasonable aspects in the original ground truth annotations. We manually re-annotated the original images to create Sharerobot-AffordV2-bench. For example, some original annotation boxes were too small, while the affordance

region referred to in the text was actually larger (as shown in Figures (a)–(d)): for the text "sit on the couch", the original ground truth region was located in the middle area, which we revised; another example is Figure (d): the text was "Indicate the parts in the figure that can be used for cutting.", but the original annotation did not cover the stem of the knife, so we also corrected this. We also evaluated models such as RoboBrain 2.0 using this revised benchmark.

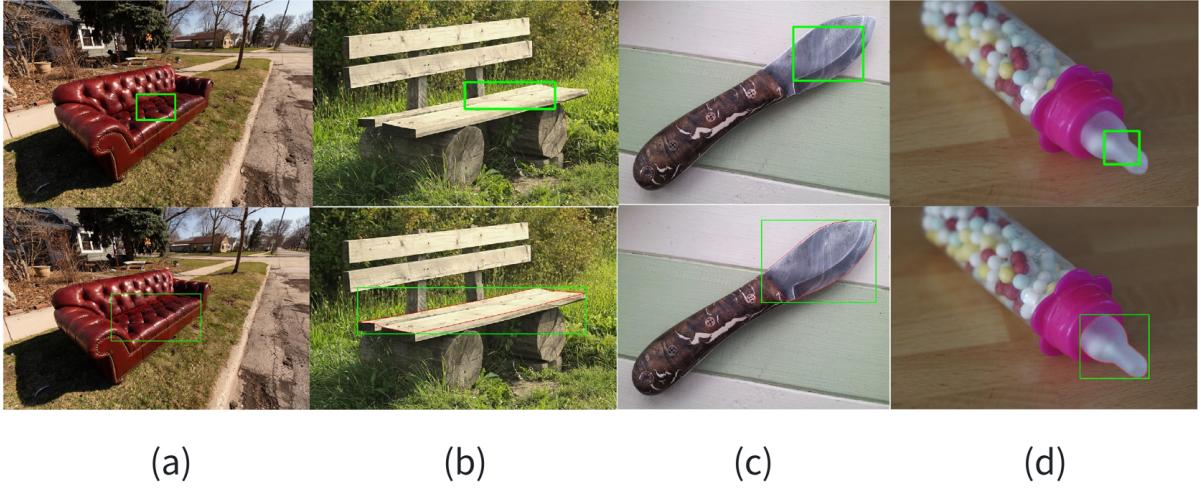


Figure 7 Partial results modified based on the ShareRobot-Bench-Affordance benchmark. The upper part shows the original ground truth results, and the lower part shows our annotated results. The area enclosed by the red line is the ground truth mask, and the green-enclosed area is the ground truth bounding box.

4.1.3 For Trajectory Evaluation

Datasets for trajectory evaluation include the trajectory dataset from Sharerobot-bench and the VABench-V dataset. We selected the trajectory data from Sharerobot-bench for evaluation, which involves judging the similarity between two trajectories. Currently, commonly used metrics include MAE (Mean Absolute Error), RMSE (Root Mean Square Error), and DFD (Dynamic Fréchet Distance). We used the DFD metric to evaluate performance on the trajectory data from Sharerobot-bench. Comparison results show that our model achieved a DFD score of 0.18, while RoboBrain 2.0-7B and RoboBrain 2.0-32B achieved 0.2368 and 0.5512, respectively. The calculation formula is as follows. Notably, we use values ranging from 1 to [the upper limit number] as the final scores, where a higher value indicates better performance.

$$\text{DFD}(P, Q) = \inf_{\gamma, \delta} \max_{t \in [0, 1]} \|P(\gamma(t)) - Q(\delta(t))\| \quad (1)$$

where:

$$\begin{aligned} P &= \{p_1, p_2, \dots, p_K\} \\ Q &= \{q_1, q_2, \dots, q_K\} \end{aligned}$$

To verify the generalization ability of the model, we established a variety of out-of-distribution evaluation scenarios, including: 1) different desktop backgrounds; 2) different containers used as placeable positions; 3) unseen objects; and 4) unseen scenarios, such as office scenarios and factory scenarios. We find that the model could correctly predict trajectories in all cases, and all 8 predicted points are correct. which indicates that the model possesses strong robustness and generalization ability.

4.2 Spatial Understanding Result

For spatial understanding and task planning, we conducted comprehensive evaluations of iFlyBot-VLM against existing embodied intelligence models and other MLLMs across six benchmarks. Specifically, the spatial understanding benchmarks include BLINK[49], EmbSpatial[50], ERQA[51], CVBench[52], and SAT[53]. The BLINK

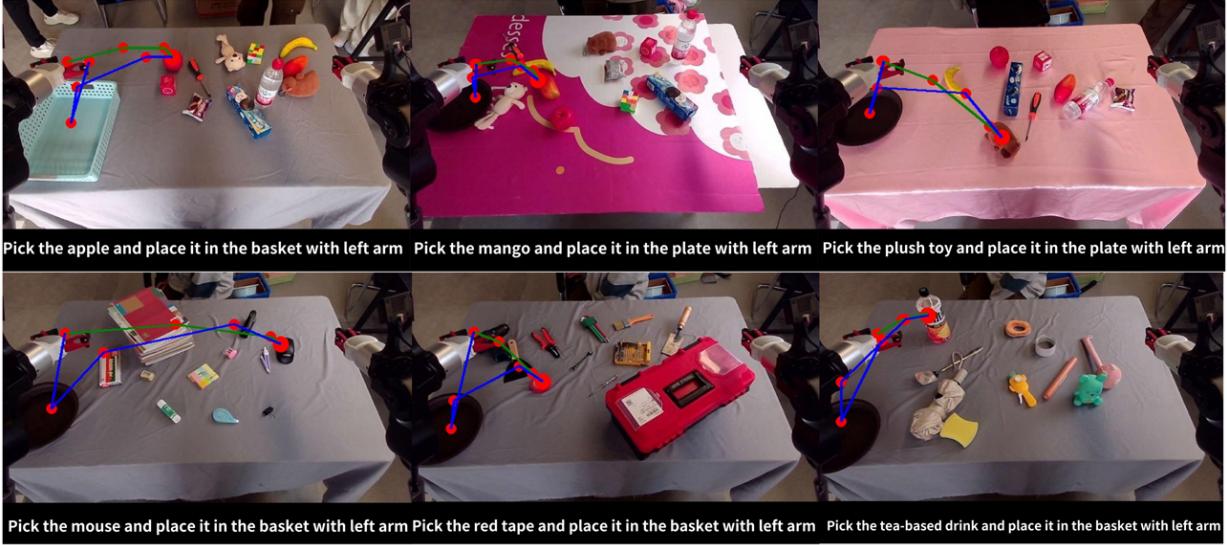


Figure 8 The generalization test results under these four unseen scenarios.

Table 2 Model Performance on Benchmarks about Spatial

| Models/Metrics | BLINK | EmbSpatial | ERQA | CVBench | SAT | EgoPlan2 |
|---------------------|-------|------------|-------|---------|-------|----------|
| Embodied-R1[44] | 66.73 | 67.4 | 39.1 | 82.7 | 70 | 30.82 |
| RoboBrain2.0-7B[45] | 83.95 | 76.32 | 34.25 | 85.75 | 75.33 | 33.23 |
| Qwen3-VL-8B[46] | 84.08 | 78.9 | 43.5 | 86.32 | 53.33 | 39.06 |
| InternVLA-M1[47] | 65.65 | 65.27 | 40.35 | 74.41 | 68 | 30.58 |
| InternVL3-8B[7] | 61.36 | 63.63 | 32.08 | 75.36 | 52.67 | 41.03 |
| iFlyBot-VLM-8B | 85.69 | 78.92 | 43.5 | 85.33 | 72 | 47 |

benchmark comprises 14 sub-tasks, among which we selected the spatially relevant components—Counting, Relative_Depth, Visual_Correspondence, and Spatial_Relation to assess the model’s capability in spatial reasoning.

For task planning, we adopted the EgoPlan2[54] benchmark, which provides a convincing measure of embodied task-planning competence. To highlight the superiority of iFlyBot-VLM in spatial understanding, we compared it against state-of-the-art embodied intelligence models that exhibit strong spatial reasoning abilities, including Embodied-R1[44], RoboBrain2.0, and InternVLA-M1. In addition, we included comparisons with open-source multimodal large models of comparable scale, such as Qwen3-VL-8B and InternVL3-8B, both of which achieve state-of-the-art performance in general multimodal understanding.

As shown in Table 2, iFlyBot-VLM achieves an accuracy of 85.69% on the BLINK benchmark, outperforming all other embodied intelligence models and open-source multimodal large models. On the EgoPlan2 benchmark, it attains the best performance with an accuracy of 47%. On ERQA, iFlyBot-VLM achieves results comparable to Qwen3-VL-8B, while significantly surpassing models such as RoboBrain2.0-7B and Embodied-R1. For CVBench, the model performs on par with Qwen3-VL-8B and RoboBrain2.0-7B, and on SAT, it achieves performance comparable to RoboBrain2.0-7B, while outperforming the current state-of-the-art model Qwen3-VL-8B.

Overall, these results demonstrate that iFlyBot-VLM exhibits strong spatial understanding capabilities, achieving consistently advanced performance across diverse evaluation benchmarks.

5 Related Work

5.1 Foundation Models

In recent years, numerous researchers and institutions have focused on the reasoning capabilities of Multimodal Large Language Models (MLLMs), such as Video-llava[55], InternVL[7, 56], Gemini 2.5[51], and Qwen3[46]. These models have demonstrated outstanding performance in general tasks such as image captioning, video understanding, and multimodal dialog. However, during training, insufficient attention is paid to spatial understanding, perception, and reasoning, resulting in inadequate spatial comprehension abilities. When applied directly to robot manipulation action generation, these models often fail.

Works including SpatialVLM[57], SpatialRGPT[58], SpatialPIN[59] and SpatialBot[60] have effectively enhanced environmental perception and interaction capabilities by strengthening models' spatial understanding. However, they also highlight the immense potential of MLLMs in embodied understanding, achieving promising results on spatial comprehension benchmarks such as BLINK[49] and SAT[53].

The aforementioned studies focus on the spatial understanding and reasoning performance of MLLMs. In embodied robot applications, the generalize ability of MLLMs alone is insufficient to achieve effective zero-shot performance on brand-new tasks. Spatial perception capabilities and embodied task planning capabilities are also key factors determining the effectiveness of Vision-Language-Action (VLA) models[27].

5.2 Datasets

Several works have proposed effective data generation and filtering solutions for spatial reasoning, providing strong support to improve the practical application of MLLMs in robotics and embodied artificial intelligence. Multi-SpatialMLLM[9] introduced the MultiSPA dataset, which contains 27 million samples. By integrating depth perception, visual correspondence, and dynamic perception, it enables MLLMs to acquire robust multi-frame spatial understanding capabilities. Meanwhile, it proposed the MultiSPA benchmark, a multi-frame spatial understanding evaluation benchmark. In the interdisciplinary field of spatial reasoning and robotics, InternSpatial[61] open-sourced a spatial reasoning dataset with 12 million question-answer pairs, covering single-view and multi-view scenarios. Enhances the spatial perception capabilities of robots by supporting supervised fine-tuning (SFT) of VLMs in robotics.

RoboRefer[23] proposed the RefSpatial dataset for spatial reference tasks, including 2.5 million samples and 20 million question-answer pairs. It not only provides training support for SFT but also caters to reinforcement fine-tuning (RFT) training. Robo2VLM[62] presented a visual question answering (VQA) dataset generation framework for vision-language models. Enhances VLMs using rich, real multimodal robot trajectory data and constructed the Robo2VLM-1 dataset, which contains 684,710 question-answer pairs. RoboBrain[48] proposed the ShareRobot dataset, which includes task planning, object affordance, and end-effector trajectories, enhancing the robot's task execution capabilities.

While the aforementioned datasets have made significant contributions to spatial reasoning in the field of embodied intelligence, they lack comprehensive data specific to the robotics domain. We further refine the robotics domain, enrich the data composition, and further enhance the model's performance in spatial understanding, perception, and reasoning.

5.3 Embodied Brains

Different from general multimodal large models, the goal of embodied VLMs is to equip models with spatial understanding and geometric perception capabilities. Researchers have proposed control parameters directly related to robot manipulation, such as points, trajectories, affordance regions, 2D grounding, and 3D grounding.

5.3.1 Functional Region Grounding

Functional region grounding is a crucial perceptual capability for robots to interact with objects. 2D grounding of individual objects cannot enable models to learn how to manipulate objects—especially objects with different part-level functions, such as tools like knives, forks, cups, and axes. Distinguishing the functional attributes of different parts of an object helps robots learn to manipulate objects more effectively. Due to the diversity of interactive objects, this task faces the challenge of establishing clear connections with object parts. [36]

proposed an affordance grounding task based on an exocentric perspective. Using only affordance labels as supervision, it learns the affordance knowledge of objects and transfers it to egocentric images. It constructed an affordance grounding dataset named AGD20K, which collects and labels over 20,000 images across 36 affordance categories and is a widely used affordance evaluation dataset.

In addition, works such as UAD[63], 2HandedAfforder[64], and Afford-X[65] are also dedicated to this field and have proposed various affordance datasets at both the part level and the entire object level—such as Handle, LVIS-Aff, and COCO-Aff—which annotate functional attributes from multiple action dimensions. Affordance-R1[66] built a high-quality reasoning dataset centered on functional affordance, named Reason Aff. Trained via reinforcement learning based on GRPO[67], it achieves strong zero-shot generalization capabilities and demonstrates emergent in-test reasoning abilities.

5.3.2 Points and Trajectories as Bridges

Many institutions have researched the use of points and trajectory points as bridges between the "brain" and action modules, achieving promising results. RoboPoint[20] proposed a vision-language model that predicts the adaptability of key points in images based on language instructions, adapting to embodied tasks. Hamster[68] fine-tuned a VLM using trajectory datasets generated from simulator and real robot datasets, and applied these trajectories as a bridge to action generation. It achieved a 20% success rate improvement in 7 task tests, demonstrating the importance of trajectories as a control parameter. MolmoAct[69] proposed a dataset containing over 10,000 high-quality robot trajectories from different scenarios and tasks. After training on this dataset, the overall performance improved by an average of 5.5% compared to the base model. A0[70] captures object-centric spatial affordance by predicting contact points and post-contact trajectories. It was pre-trained on 1 million contact point data and fine-tuned on annotated trajectories. After mapping the inferred trajectory data to 3D space, it achieved state-of-the-art (SOTA) results compared to RDT in wiping tasks. In addition, Magma[71] has also achieved notable results in the use of points.

5.3.3 Comprehensive Models

Some comprehensive models, such as Embodied-FSD, leverage the visual understanding capabilities of VLMs. They use data such as spatial affordance boxes/points and visual trajectories to provide expressive and compact spatial information. By introducing the FSD model and the SrCOT method, they achieve spatial understanding and reasoning in embodied tasks, and proposed the FSD dataset. After fine-tuning on some open-source models using this data, the average success rate of task execution increased by more than 10%. In addition, 2D trajectories inferred based on COT are mapped to 3D space, and complex tasks such as cloth folding are completed using Curobo for operation execution.

RoboBrain enhances robot manipulation capabilities by constructing ShareRobot data and general multimodal data, and adopting a multi-stage training strategy. Embodied-R1[44] uses "pointing" as an intermediate state to connect high-level vision-language understanding and low-level actions. Through two-stage RFT training, it builds the model's spatial reasoning capabilities and multi-task embodied pointing capabilities. RoboBrain-2.0[45] adopts a multi-stage training strategy, enabling the model to support spatial understanding, temporal modeling, and chain-of-thought reasoning in embodied environments. InternVLA-M1[47] was pre-trained on over 2.3 million spatial reasoning data samples, taking spatial localization as a key link in vision-language-action training, highlighting that spatially guided training can scale the embodied capabilities of general robots.

Besides, 3D grounding capability is also an important control parameter for embodied brains, which is more conducive to models learning spatial relationships and sizes between objects. For example, Qwen3-VL and Gemini 2.0 have conducted research in this area.

These works promote the advancement of embodied foundation models from the 2D world to the 3D world. The aforementioned solutions have made significant breakthroughs in spatial understanding, reasoning, and task planning. However, the continuous improvement of robot embodied tasks requires robots to possess certain general thinking and reasoning capabilities. Based on this, this paper further refines spatial reasoning tasks to enhance the spatial understanding capabilities of MLLMs. Additionally, by constructing rich multimodal data,

we improve the spatial reasoning, understanding, and task planning capabilities of MLLMs while equipping them with general image-text reasoning capabilities.

6 Conclusion & Future Work

In this report, we introduce our vision-language foundation model designed for embodied scenarios, and elaborate on the dataset creation method, model architecture, and evaluation approach in detail. We have open-sourced the model checkpoints, datasets, and evaluation tools. Our model possesses spatial understanding and reasoning capabilities, task planning capabilities, as well as spatial perception capabilities including spatial pointing, affordance regions, 2D trajectories, object grasping poses, and 3D bounding boxes (3Dbox). Compared with general multimodal models and the latest industry-leading embodied multimodal models on typical evaluation datasets, our model has achieved outstanding performance.

In future work, we will continue to enhance the model's spatial understanding and spatial perception capabilities, and densely integrate them into Vision-Language-Action (VLA) models for application. We will focus on the following key directions:

1. Incorporating image generation capabilities into the model, and introducing a world model to enable the model to learn future prediction. This will enhance spatial understanding capabilities and yield more accurate 2D and 3D trajectories.
2. Expanding the model's self-reflection ability and multimodal input-output capabilities such as scene reconstruction, point clouds, and 3D trajectories.
3. Applying this model to the VLA framework to improve the model's generalization ability and step-by-step execution capability for long-horizon tasks.

References

- [1] Z. Yi, T. Xiao, and M. V. Albert, “A survey on multimodal large language models in radiology for report generation and visual question answering,” *Information*, vol. 16, no. 2, p. 136, 2025.
- [2] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, “A comprehensive overview of large language models,” *ACM Transactions on Intelligent Systems and Technology*, vol. 16, no. 5, pp. 1–72, 2025.
- [3] K. Kawaharazuka, J. Oh, J. Yamada, I. Posner, and Y. Zhu, “Vision-language-action models for robotics: A review towards real-world applications,” *IEEE Access*, 2025.
- [4] R. Shao, W. Li, L. Zhang, R. Zhang, Z. Liu, R. Chen, and L. Nie, “Large vlm-based vision-language-action models for robotic manipulation: A survey,” *arXiv preprint arXiv:2508.13073*, 2025.
- [5] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter et al., “ π_0 : A vision-language-action flow model for general robot control,” *arXiv preprint arXiv:2410.24164*, 2024.
- [6] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai et al., “ $\pi_{0.5}$: a vision-language-action model with open-world generalization,” *arXiv preprint arXiv:2504.16054*, 2025.
- [7] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu et al., “Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 24185–24198.
- [8] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, “Scannet: Richly-annotated 3d reconstructions of indoor scenes,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.
- [9] R. Xu, W. Wang, H. Tang, X. Chen, X. Wang, F.-J. Chu, D. Lin, M. Feiszli, and K. J. Liang, “Multi-spatialmlm: Multi-frame spatial understanding with multi-modal large language models,” *arXiv preprint arXiv:2505.17015*, 2025.
- [10] A. W. C. contributors, “Agibot world colosseum,” <https://github.com/OpenDriveLab/AgiBot-World>, 2024.

- [11] K. Wu, C. Hou, J. Liu, Z. Che, X. Ju, Z. Yang, M. Li, Y. Zhao, Z. Xu, G. Yang et al., “Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation,” in *Robotics: Science and Systems (RSS) 2025*. Robotics: Science and Systems Foundation, 2025. [Online]. Available: <https://www.roboticsproceedings.org/rss21/p152.pdf>
- [12] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, “Depth anything: Unleashing the power of large-scale unlabeled data,” in *CVPR*, 2024.
- [13] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini et al., “Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models,” *arXiv e-prints*, pp. arXiv-2409, 2024.
- [14] M. Acharya, K. Kafle, and C. Kanan, “Tallyqa: Answering complex counting questions,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 8076–8084.
- [15] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov et al., “The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale,” *International journal of computer vision*, vol. 128, no. 7, pp. 1956–1981, 2020.
- [16] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma et al., “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” *International journal of computer vision*, vol. 123, no. 1, pp. 32–73, 2017.
- [17] R. Khanam and M. Hussain, “Yolov11: An overview of the key architectural enhancements,” *arXiv preprint arXiv:2410.17725*, 2024.
- [18] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blstein, O. Ram, D. Zhang, E. Rosen et al., “Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities,” *arXiv preprint arXiv:2507.06261*, 2025.
- [19] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, “Modeling context in referring expressions,” in *European conference on computer vision*. Springer, 2016, pp. 69–85.
- [20] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox, “Robopoint: A vision-language model for spatial affordance prediction for robotics,” *arXiv preprint arXiv:2406.10721*, 2024.
- [21] H. R. Walké, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du et al., “Bridgedata v2: A dataset for robot learning at scale,” in *Conference on Robot Learning*. PMLR, 2023, pp. 1723–1736.
- [22] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su et al., “Grounding dino: Marrying dino with grounded pre-training for open-set object detection,” in *European conference on computer vision*. Springer, 2024, pp. 38–55.
- [23] E. Zhou, J. An, C. Chi, Y. Han, S. Rong, C. Zhang, P. Wang, Z. Wang, T. Huang, L. Sheng et al., “Roborefer: Towards spatial referring with reasoning in vision-language models for robotics,” *arXiv preprint arXiv:2506.04308*, 2025.
- [24] Z. Luo, Y. Yang, Y. Zhang, and F. Zheng, “Roboreflect: A robotic reflective reasoning framework for grasping ambiguous-condition objects,” *arXiv preprint arXiv:2501.09307*, 2025.
- [25] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis et al., “Droid: A large-scale in-the-wild robot manipulation dataset,” *arXiv preprint arXiv:2403.12945*, 2024.
- [26] G. Baruch, Z. Chen, A. Dehghan, T. Dimry, Y. Feigin, P. Fu, T. Gebauer, B. Joffe, D. Kurz, A. Schwartz et al., “Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data,” *arXiv preprint arXiv:2111.08897*, 2021.
- [27] Y. Yuan, H. Cui, Y. Chen, Z. Dong, F. Ni, L. Kou, J. Liu, P. Li, Y. Zheng, and J. Hao, “From seeing to doing: Bridging reasoning and decision for robotic manipulation,” *arXiv preprint arXiv:2505.08548*, 2025.
- [28] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan et al., “Grounded sam: Assembling open-world models for diverse visual tasks,” *arXiv preprint arXiv:2401.14159*, 2024.

- [29] A. Guo, B. Wen, J. Yuan, J. Tremblay, S. Tyree, J. Smith, and S. Birchfield, “Handal: A dataset of real-world manipulable object categories with pose annotations, affordances, and reconstructions,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 11 428–11 435.
- [30] Z. Wan, Y. Xie, C. Zhang, Z. Lin, Z. Wang, S. Stepputtis, D. Ramanan, and K. P. Sycara, “Instructpart: Affordance-based part segmentation from language instruction,” in *AAAI-2024 Workshop on Public Sector LLMs: Algorithmic and Sociotechnical Design*, 2024.
- [31] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, “Affordance detection of tool parts from geometric features,” in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 1374–1381.
- [32] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, “Object-based affordances detection with convolutional neural networks and dense conditional random fields,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [33] V. Ramanathan, A. Kalia, V. Petrovic, Y. Wen, B. Zheng, B. Guo, R. Wang, A. Marquez, R. Kovvuri, A. Kadian et al., “Paco: Parts and attributes of common objects,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7141–7151.
- [34] A. Gupta, P. Dollar, and R. Girshick, “Lvis: A dataset for large vocabulary instance segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5356–5364.
- [35] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu et al., “Ego4d: Around the world in 3,000 hours of egocentric video,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 995–19 012.
- [36] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao, “Learning affordance grounding from exocentric images,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 2252–2261.
- [37] A. O’Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain et al., “Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6892–6903.
- [38] D. Niu, Y. Sharma, G. Biamby, J. Quenum, Y. Bai, B. Shi, T. Darrell, and R. Herzig, “Llarva: Vision-action instruction tuning enhances robot learning,” *arXiv preprint arXiv:2406.11815*, 2024.
- [39] J. Lazarow, D. Griffiths, G. Kohavi, F. Crespo, and A. Dehghan, “Cubify anything: Scaling indoor 3d object detection,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 22 225–22 233.
- [40] S. Song, S. P. Lichtenberg, and J. Xiao, “Sun rgb-d: A rgb-d scene understanding benchmark suite,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 567–576.
- [41] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann, “Objectron: A large scale dataset of object-centric videos in the wild with pose annotations,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7822–7831.
- [42] Y. Chen, Y. Ge, Y. Ge, M. Ding, B. Li, R. Wang, R. Xu, Y. Shan, and X. Liu, “Egoplan-bench: Benchmarking egocentric embodied planning with multimodal large language models,” *CoRR*, 2023.
- [43] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price et al., “Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100,” *International Journal of Computer Vision*, vol. 130, no. 1, pp. 33–55, 2022.
- [44] Y. Yuan, H. Cui, Y. Huang, Y. Chen, F. Ni, Z. Dong, P. Li, Y. Zheng, and J. Hao, “Embodied-r1: Reinforced embodied reasoning for general robotic manipulation,” *arXiv preprint arXiv:2508.13998*, 2025.
- [45] B. R. Team, M. Cao, H. Tan, Y. Ji, X. Chen, M. Lin, Z. Li, Z. Cao, P. Wang, E. Zhou et al., “Robobrain 2.0 technical report,” *arXiv preprint arXiv:2507.02029*, 2025.
- [46] Q. Team, “Qwen3 technical report,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.09388>
- [47] X. Chen, Y. Chen, Y. Fu, N. Gao, J. Jia, W. Jin, H. Li, Y. Mu, J. Pang, Y. Qiao et al., “Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy,” *arXiv preprint arXiv:2510.13778*, 2025.
- [48] Y. Ji, H. Tan, J. Shi, X. Hao, Y. Zhang, H. Zhang, P. Wang, M. Zhao, Y. Mu, P. An et al., “Robobrain: A unified brain model for robotic manipulation from abstract to concrete,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1724–1734.

- [49] X. Fu, Y. Hu, B. Li, Y. Feng, H. Wang, X. Lin, D. Roth, N. A. Smith, W.-C. Ma, and R. Krishna, “Blink: Multimodal large language models can see but not perceive,” in *European Conference on Computer Vision*. Springer, 2024, pp. 148–166.
- [50] M. Du, B. Wu, Z. Li, X. Huang, and Z. Wei, “Embspatial-bench: Benchmarking spatial understanding for embodied tasks with large vision-language models,” 2024. [Online]. Available: <https://arxiv.org/abs/2406.05756>
- [51] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl et al., “Gemini robotics: Bringing ai into the physical world,” *arXiv preprint arXiv:2503.20020*, 2025.
- [52] P. Tong, E. Brown, P. Wu, S. Woo, A. J. V. IYER, S. C. Akula, S. Yang, J. Yang, M. Middepogu, Z. Wang et al., “Cambrian-1: A fully open, vision-centric exploration of multimodal llms,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 87310–87356, 2024.
- [53] A. Ray, J. Duan, R. Tan, D. Bashkirova, R. Hendrix, K. Ehsani, A. Kembhavi, B. A. Plummer, R. Krishna, K.-H. Zeng et al., “Sat: Spatial aptitude training for multimodal language models,” *arXiv e-prints*, pp. arXiv–2412, 2024.
- [54] L. Qiu, Y. Chen, Y. Ge, Y. Ge, Y. Shan, and X. Liu, “Egoplan-bench2: A benchmark for multimodal large language model planning in real-world scenarios,” *arXiv preprint arXiv:2412.04447*, 2024.
- [55] B. Lin, Y. Ye, B. Zhu, J. Cui, M. Ning, P. Jin, and L. Yuan, “Video-llava: Learning united visual representation by alignment before projection,” *arXiv preprint arXiv:2311.10122*, 2023.
- [56] Z. Chen, W. Wang, Y. Cao, Y. Liu, Z. Gao, E. Cui, J. Zhu, S. Ye, H. Tian, Z. Liu et al., “Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling,” *arXiv preprint arXiv:2412.05271*, 2024.
- [57] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia, “Spatialvlm: Endowing vision-language models with spatial reasoning capabilities,” in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2024, pp. 14455–14465.
- [58] A.-C. Cheng, H. Yin, Y. Fu, Q. Guo, R. Yang, J. Kautz, X. Wang, and S. Liu, “Spatialrgpt: Grounded spatial reasoning in vision-language models,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 135 062–135 093, 2024.
- [59] C. Ma, K. Lu, T.-Y. Cheng, N. Trigoni, and A. Markham, “Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors,” *Advances in neural information processing systems*, vol. 37, pp. 68 803–68 832, 2024.
- [60] W. Cai, I. Ponomarenko, J. Yuan, X. Li, W. Yang, H. Dong, and B. Zhao, “Spatialbot: Precise spatial understanding with vision language models,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 9490–9498.
- [61] N. Deng, L. Gu, S. Ye, Y. He, Z. Chen, S. Li, H. Wang, X. Wei, T. Yang, M. Dou et al., “Internspatial: A comprehensive dataset for spatial reasoning in vision-language models,” *arXiv preprint arXiv:2506.18385*, 2025.
- [62] K. Chen, S. Xie, Z. Ma, P. R. Sanketi, and K. Goldberg, “Robo2vilm: Visual question answering from large-scale in-the-wild robot manipulation datasets,” *arXiv preprint arXiv:2505.15517*, 2025.
- [63] Y. Tang, W. Huang, Y. Wang, C. Li, R. Yuan, R. Zhang, J. Wu, and L. Fei-Fei, “Uad: Unsupervised affordance distillation for generalization in robotic manipulation,” *arXiv preprint arXiv:2506.09284*, 2025.
- [64] M. Heidinger, S. Jauhri, V. Prasad, and G. Chalvatzaki, “2handedafforder: Learning precise actionable bimanual affordances from human videos,” *arXiv preprint arXiv:2503.09320*, 2025.
- [65] X. Zhu, Y. Li, L. Cui, P. Li, H.-a. Gao, Y. Zhu, and H. Zhao, “Afford-x: Generalizable and slim affordance reasoning for task-oriented manipulation,” *arXiv preprint arXiv:2503.03556*, 2025.
- [66] H. Wang, S. Wang, Y. Zhong, Z. Yang, J. Wang, Z. Cui, J. Yuan, Y. Han, M. Liu, and Y. Ma, “Affordance-r1: Reinforcement learning for generalizable affordance reasoning in multimodal large language model,” *arXiv preprint arXiv:2508.06206*, 2025.
- [67] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi et al., “Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning,” *arXiv preprint arXiv:2501.12948*, 2025.

- [68] Y. Li, Y. Deng, J. Zhang, M. Memmel, R. Yu, C. R. Garrett, F. Ramos, D. Fox, A. Li et al., "Hamster: Hierarchical action models for open-world robot manipulation," [arXiv preprint arXiv:2502.05485](#), 2025.
- [69] J. Lee, J. Duan, H. Fang, Y. Deng, S. Liu, B. Li, B. Fang, J. Zhang, Y. R. Wang, S. Lee et al., "Molmoact: Action reasoning models that can reason in space," [arXiv preprint arXiv:2508.07917](#), 2025.
- [70] R. Xu, J. Zhang, M. Guo, Y. Wen, H. Yang, M. Lin, J. Huang, Z. Li, K. Zhang, L. Wang et al., "A0: An affordance-aware hierarchical model for general robotic manipulation," [arXiv preprint arXiv:2504.12636](#), 2025.
- [71] L. Lu, W. Zhai, H. Luo, Y. Kang, and Y. Cao, "Phrase-based affordance detection via cyclic bilateral interaction," [IEEE Transactions on Artificial Intelligence](#), vol. 4, no. 5, pp. 1186–1198, 2022.

7 Appendix

7.1 Appendix A

7.1.1 Prompt for Chain-of-Thought (CoT) on Affordance

You are a senior data annotation expert. Your task is to analyze image information based on the given image and User Query, and generate a detailed English thinking process to identify the **affordance area** of the target object from a robotic perspective that enables the specified task. Please strictly follow the below task requirements and thinking steps for your output.

Task Requirements

1. Understand the User Query and image content. The User Query will clearly specify an object and a task that can be completed. The image provides visual information, with a red rectangular box marking the corresponding affordance area. 2. Develop a logical English thinking process. This process must mimic the "reasoning" of a Vision-Language Model (VLM): starting from the image content and user query, and gradually deducing which area is the affordance area for completing the specified task. Your thinking process must align with the position of the affordance area and the object mentioned in the User Query. 3. Adhere to key constraints: Do not reveal the fact that the image has manual annotations. Your thinking process must be independent and derived from "observation" and "understanding" of the image itself. You must not state phrases like "there is already a red box on the image" or "this box is pre-annotated". Instead, you need to describe the shape, position of this area, and the physical properties that enable it to complete the task using descriptive language. Additionally, neither your thinking process nor output content shall include any coordinate values.

Thinking Steps

1. What is the current task? Analysis: The task requires identifying or locating which part of the object serves a specific purpose. This requires understanding the object's function and combining it with image information for judgment.
2. Where is the target object in the image, and what is its current posture (e.g., upright, lying flat, occluded) or state (e.g., microwave is closed, gripper is open)? Analysis: Describe the position and state of the target object in the image to lay the foundation for subsequent functional reasoning.
3. What are the common operable areas of this object? Analysis: Based on common-sense understanding of the object, deduce which area is typically used when performing the specific task.
4. If it is a grasping task, is the target within the grasping range or is there sufficient operating space? Analysis: Determine whether the task is a grasping task or a functional area identification task. If there is a clear robotic arm or gripper in the image, it is a grasping task—evaluate whether the operable area is accessible and has sufficient space. This step does not apply if it is not a grasping task.
5. Finally, summarize and identify which operable area can enable the entire task. Analysis: Conclude and confirm that, based on the above analysis, the identified area is indeed the correct choice for completing the task.

Provided User Query:

Example 1:

User Query: Please indicate a possible affordance area of the baseball bat that can be used for holding

Output: First, I need to determine the current task. The task is to identify the functional area on a baseball bat that is used for "holding." This requires understanding the bat's purpose as a tool and recognizing the specific part designed for gripping. Next, I need to analyze the object's location, posture, and current state within the image. In the picture, the baseball bat is on the left side, positioned vertically with no obvious obstructions. Then, I'll analyze the common operable areas of this object. Based on common knowledge, a baseball bat is used for hitting a ball. To swing effectively, a person typically holds the narrower, bottom part of the bat, which is the handle. This handle area is specifically designed for gripping. I must also determine if this is a grasping task and if the object is within grasping range or has sufficient operating space. Based on my analysis, this is not a grasping task but a "functional area identification" task. Therefore, grasping range and operating space are not applicable. Finally, I need to summarize and point out which operable area can accomplish the entire task. By holding the bat by its handle, a person can gain a stable grip and control it for the subsequent swinging motion. Therefore, the handle is the correct area on the baseball bat to complete this task.

Example 2:

User Query: <Imagehere> Please indicate a possible affordance area of the bench that can be used for sitting

Output: First, I need to determine the current task. The task is to identify the functional area on a bench that is used for "sitting." This requires understanding the function of a bench as a piece of furniture that provides a resting area for people or robots. Next, I need to analyze the object's location, posture, and current state within the image. There is a bench in the center of the frame. It's positioned horizontally and steadily, with no obvious obstructions. Its structure includes a seating surface and four support legs. Then, I will analyze the common operable areas of this object. The primary function of a bench is to provide a place to sit. Therefore, its horizontal seating surface is the specific operable area designed for this purpose. This area is typically flat and stable to support the weight of a person or object. I must also determine if this is a grasping task and if the object is within grasping range or has sufficient operating space. Based on my analysis, this is not a grasping task, but a "functional area identification" task. I need to understand and recognize the area's "affordance" for sitting. Therefore, evaluating grasping range or operating space is not necessary. Finally, I need to summarize and point out which operable area can accomplish the entire task. Based on the analysis above, the seating surface of the bench is the affordance area for sitting. Locating and indicating this area completes the task.

7.2 Appendix B

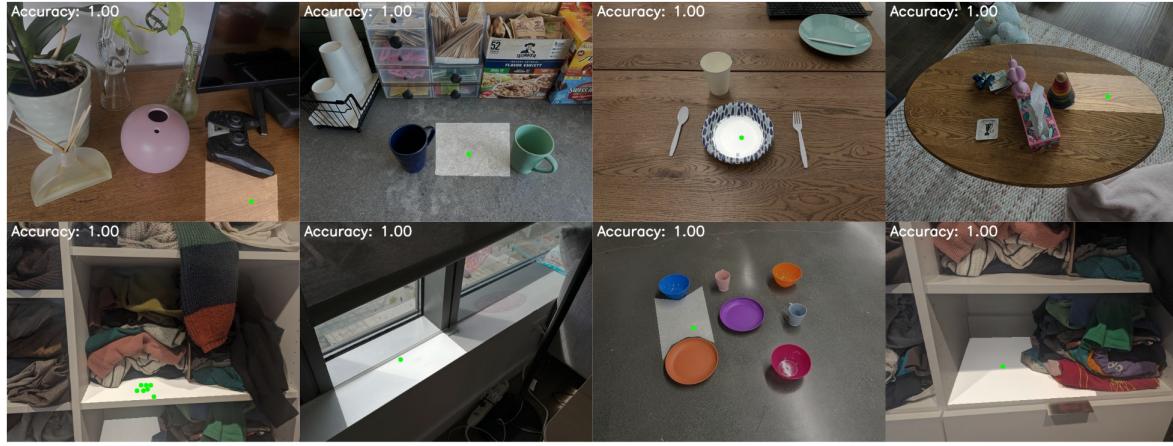


Figure 9 Partial results on the Where2Place-bench.

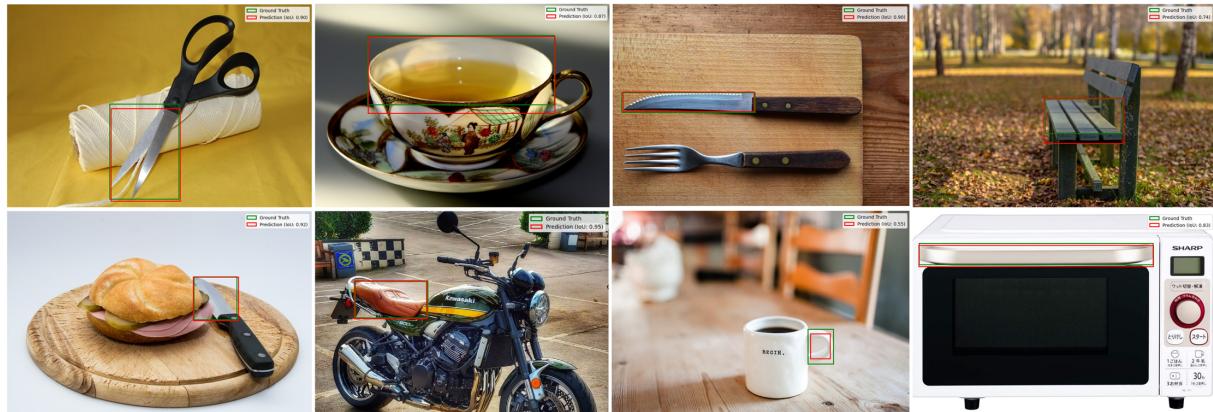


Figure 10 Partial results on the modified Sharerobot-bench.

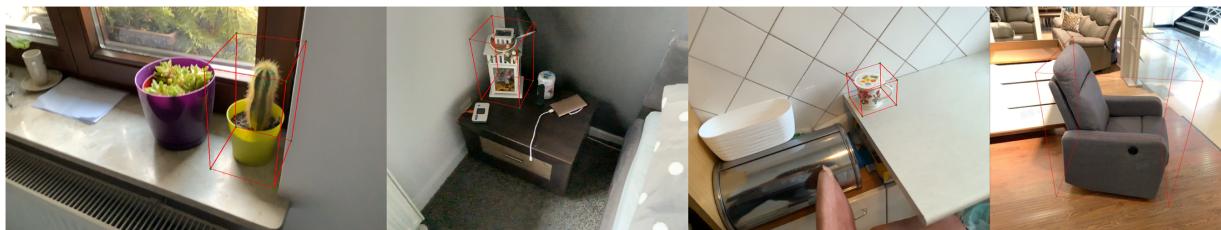


Figure 11 Partial results on the iFlyBot-3DGounding-bench.



Figure 12 Partial results on the Sharerobot-bench.



Figure 13 Partial results on the iFlyBot-GraspPose-bench.

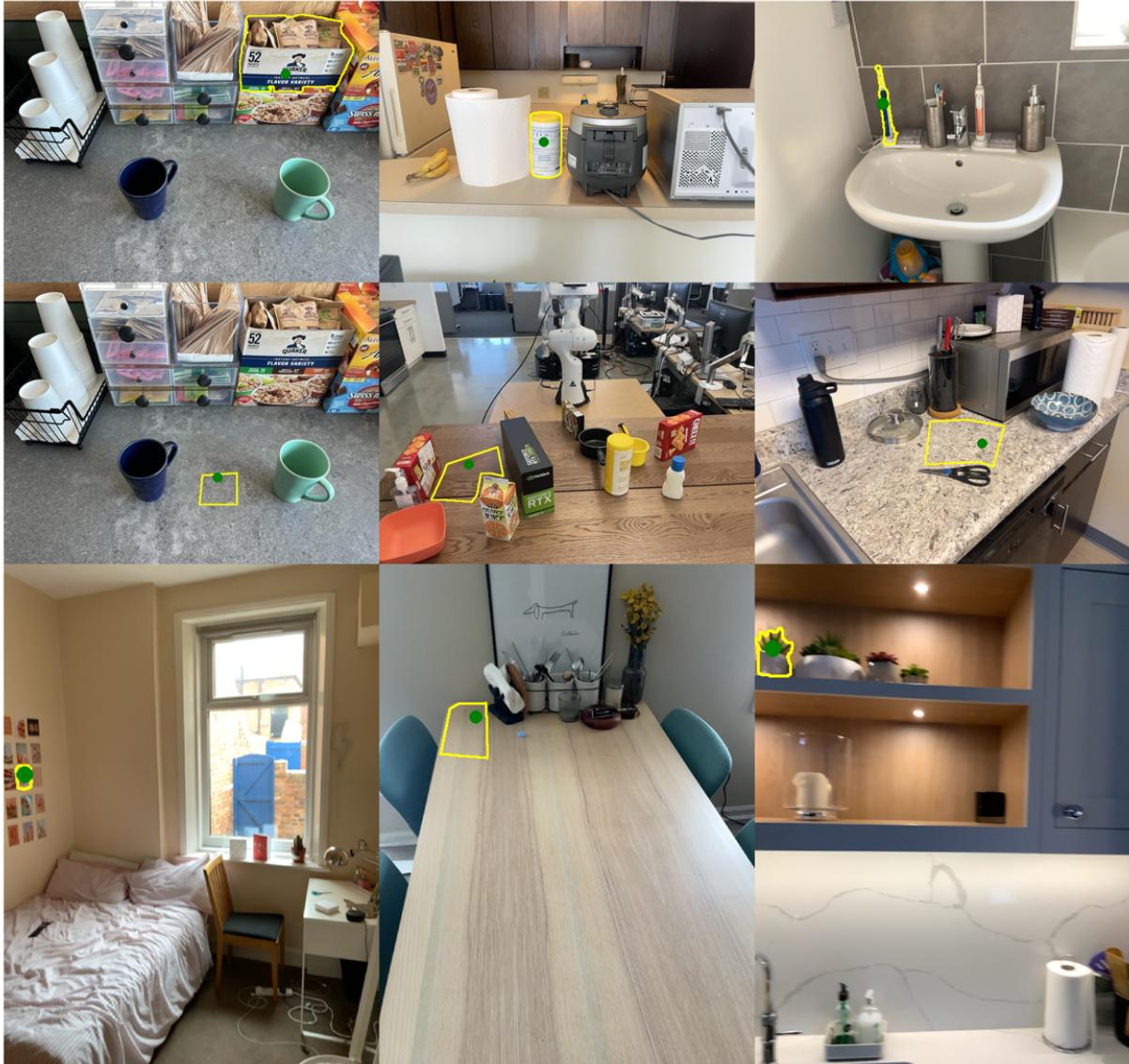


Figure 14 Partial results on the RefSpatial-bench.